DOCKET NO.: 05-03-010
CLIENT NO.: UGSC01-05022
Customer No.: 45113

PATENT

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re application of	:	Graham Hughes, et al.
Application No.	:	10/706,848
Filed	:	November 12, 2003
For	:	SYSTEM, METHOD, AND COMPUTER PROGRAM PRODUCT FOR DISTRIBUTED TESTING OF PROGRAM CODE
Group No.	:	2192
Examiner	:	Eric B. Kiss
Conf. No.	:	6082

Mail Stop Appeal Brief - Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Sir:

CERTIFICATE OF MAILING BY FIRST CLASS MAIL

The undersigned hereby certifies that the following documents:

1. Postcard receipt;
2. Appeal Brief;
3. Check in the amount of $510.00 for Appeal Brief filing fee; and
4. Fee Transmittal FY 2008

relating to the above application, were deposited as "First Class Mail" with the United States Postal Service, addressed to Mail Stop Appeal Brief - Patents, Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450, on September 22, 2008.

Date: 09-22-08

April Morgan
Mailer

Date: 09-22-08

Matthew S. Anderson
Reg. No. 39,093

P.O. Drawer 800889
Dallas, Texas 75380
Phone: (972) 628-3600
Fax: (972) 628-3616
E-mail: *manderson@munckcarter.com*

PTO/SB/17 (10-07)
Approved for use through 06/30/2010. OMB 0651-0032
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995 no persons are required to respond to a collection of information unless it displays a valid OMB control number

Effective on 12/08/2004.
Fees pursuant to the Consolidated Appropriations Act, 2005 (H.R. 4818).

FEE TRANSMITTAL
For FY 2008

□ Applicant claims small entity status. See 37 CFR 1.27

TOTAL AMOUNT OF PAYMENT ($) 510.00

Complete if Known	
Application Number	10/706,848
Filing Date	November 12, 2003
First Named Inventor	Graham Hughes
Examiner Name	Eric B. Kiss
Art Unit	2192
Attorney Docket No.	05-03-010 (UGSC01-05022)

METHOD OF PAYMENT (check all that apply)

☑ Check □ Credit Card □ Money Order □ None □ Other (please identify): ____________

☑ Deposit Account Deposit Account Number: 50-0208 Deposit Account Name: Munck Carter, P.C.

For the above-identified deposit account, the Director is hereby authorized to: (check all that apply)

□ Charge fee(s) indicated below □ Charge fee(s) indicated below, **except for the filing fee**

☑ Charge any additional fee(s) or underpayments of fee(s) under 37 CFR 1.16 and 1.17 ☑ Credit any overpayments

WARNING: Information on this form may become public. Credit card information should not be included on this form. Provide credit card information and authorization on PTO-2038.

FEE CALCULATION

1. BASIC FILING, SEARCH, AND EXAMINATION FEES

	FILING FEES		SEARCH FEES		EXAMINATION FEES		
Application Type	Fee ($)	Small Entity Fee ($)	Fee ($)	Small Entity Fee ($)	Fee ($)	Small Entity Fee ($)	Fees Paid ($)
Utility	310	155	510	255	210	105	______
Design	210	105	100	50	130	65	______
Plant	210	105	310	155	160	80	______
Reissue	310	155	510	255	620	310	______
Provisional	210	105	0	0	0	0	______

2. EXCESS CLAIM FEES

Fee Description	Fee ($)	Small Entity Fee ($)
Each claim over 20 (including Reissues)	50	25
Each independent claim over 3 (including Reissues)	210	105
Multiple dependent claims	370	185

Total Claims Extra Claims Fee ($) Fee Paid ($)
______ - 20 or HP = ______ x ______ = ______
HP = highest number of total claims paid for, if greater than 20.

Indep. Claims Extra Claims Fee ($) Fee Paid ($)
______ - 3 or HP = ______ x ______ = ______
HP = highest number of independent claims paid for, if greater than 3.

Multiple Dependent Claims
Fee ($) ______ Fee Paid ($) ______

3. APPLICATION SIZE FEE

If the specification and drawings exceed 100 sheets of paper (excluding electronically filed sequence or computer listings under 37 CFR 1.52(e)), the application size fee due is $260 ($130 for small entity) for each additional 50 sheets or fraction thereof. See 35 U.S.C. 41(a)(1)(G) and 37 CFR 1.16(s).

Total Sheets Extra Sheets Number of each additional 50 or fraction thereof Fee ($) Fee Paid ($)
______ - 100 = ______ / 50 = ______ (round up to a whole number) x ______ = ______

4. OTHER FEE(S) Fees Paid ($)

Non-English Specification, $130 fee (no small entity discount) ______

Other (e.g., late filing surcharge): Appeal Brief filing fee $510.00

SUBMITTED BY

Signature		Registration No. (Attorney/Agent) 39,093	Telephone 972-628-3600
Name (Print/Type)	Matthew S. Anderson		Date September 22, 2008

This collection of information is required by 37 CFR 1.136. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 30 minutes to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. **SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.**

If you need assistance in completing the form, call 1-800-PTO-9199 and select option 2.

DOCKET NO.: 05-03-010 (UGSC01-05022)
Customer No. 45113

PATENT

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of	:	Graham Hughes, et al.
Serial No.	:	10/706,848
Filed	:	November 12, 2003
For	:	SYSTEM, METHOD, AND COMPUTER PROGRAM PRODUCT FOR DISTRIBUTED TESTING OF PROGRAM CODE
Group No.	:	2192
Examiner	:	Eric B. Kiss
Conf. No.	:	6082

MAIL STOP APPEAL BRIEF - PATENTS
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

APPEAL BRIEF

Sir:

Applicants herewith respectfully submit that the Examiner's decision of March 21, 2008, finally rejecting Claims 1-21 in the present application, should be reversed, in view of the following arguments and authorities. This Brief is submitted in response to the Notice of Panel Decision mailed August 11, 2008. A check in the amount of $510.00 is enclosed for the fee of filing a Brief on Appeal, but please charge any additional necessary fees to Deposit Account No. 50-0208.

TABLE OF CONTENTS

Table of Authorities

Cases

Regulations

Real Party in Interest

The real party in interest, and assignee of this case, is Siemens Product Lifecycle Management Software Inc.

Related Appeals or Interferences

To the best knowledge and belief of the undersigned attorney, there are none.

Status of Claims

Claims 1-21 are under final rejection, and are each appealed.

Status of Amendments after Final

Claims 8 and 11 were amended after final rejection to correct informalities, and this amendment was entered. No other claims were amended after final rejection.

Summary of Claimed Subject Matter

The following summary refers to disclosed embodiments and their advantages, but does not delimit any of the claimed inventions.

In General

The present application is directed, in general, to software development and testing. *Page 1, lines 5-6.*

Support for Independent Claims

Note that, per 37 CFR §41.37, only each of the independent claims are discussed in this section, as well as any claims including means-plus-function language that is argued separately below. In the arguments below, however, the dependent claims are also discussed and distinguished from the prior art. The discussion of the claims is for illustrative purposes, and is not intended to affect the scope of the claims.

Claim 1 describes a method for testing code (*e.g., as illustrated in the flow diagram of Fig. 3*). The method includes a receiving a test request (*e.g., at block 305*). The method also includes sending executable program code (*e.g., at block 315*), corresponding to the test request, to a client system (*e.g., any of clients 215/220/225/230 illustrated in Fig. 2*). The method also includes receiving a response from the client system (*e.g., at block 320*) indicating that the client system will perform a test, and indicating that the client system was not being actively used when the executable program code was sent. (*Application, Page 9, Line 15 – Page 10, Line 17; Page 10, Line 18 – Page 12, Line 3; and Figures 2 - 4).*

Claim 4 describes a method for testing code in a client data processing system (*e.g., as illustrated in the flow diagram of Fig. 4*). The method includes receiving executable code (*e.g., at block 405*) from a server system (*e.g., server 205 illustrated in Fig. 2*) in a client data processing system (*e.g., any of clients 215/220/225/230 illustrated in Fig. 2*). The method describes if the client data processing system is in an idle state when the executable code is received (*e.g., at block 410*), then sending a response to the server system (*e.g., at block 420*), testing at least a portion of the executable code (*e.g., at block 425*), and sending test results to the server system (*e.g., at block 430*). (*Application, Page 9, Line 15 – Page 10, Line 17; Page 10, Line 18 – Page 12, Line 3; and Figures 2 - 4).*

Claim 8 is similar to claim 1, and describes a data processing system (*e.g., as illustrated in Fig. 1*) comprising a processor (*e.g., processor 102*) and accessible memory (*e.g., memory 108*). The data processing system is configured to receive a test request (*e.g., at block 305*). The data processing system is also configured to send executable program code (*e.g., at block 315*), corresponding to the test request, to a client system (*e.g., any of clients 215/220/225/230 illustrated in Fig. 2*). The data processing system is also configured to receive a response from the client system indicating that the client system will perform a test, and indicating that the client system was not being actively used when the executable program code was sent (*e.g., at block 320*). (*Application, Page 6, Line 19 – Page 8, Line 8; Page 9, Line 15 – Page 10, Line 17; Page 10, Line 18 – Page 12,* *Line 3; and Figures 1 - 4).*

Claim 11 is similar to claim 4, and describes a data processing system (*e.g., as illustrated in Fig. 1*) comprising a processor (*e.g., processor 102*) and accessible memory (*e.g., memory 108*). The data processing system is configured to receive executable code from a server system in a client data processing system. The data processing system is configured so that if the client data processing system is in an idle state when the executable code is received (*e.g., at block 410*), it will send a response to the server system (*e.g., at block 420*), test at least a portion of the executable code (*e.g., at block 425*), and send test results to the server system (*e.g., at block 430*). (*Application, Page 6, Line 19 – Page 8, Line 8; Page 9, Line 15 – Page 10, Line 17; Page 10, Line 18 – Page 12, Line 3; and Figures 1 - 4).*

Claim 15 is similar to claim 1, and describes a computer program product tangibly embodied in a machine-readable medium (*e.g., memory 108*). The computer program product includes instructions for receiving a test request (*e.g., at block 305*). The computer program product also includes instructions for sending executable program code (*e.g., at block 315*), corresponding to the test request, to a client system (*e.g., any of clients 215/220/225/230 illustrated in Fig. 2*). The computer program product also includes instructions for receiving a response from the client system (*e.g., at block 320*) indicating that the client system will perform a test, and indicating that the client system was not being actively used when the executable program code was sent. (*Application, Page 9, Line 15 – Page 10, Line 17; Page 10, Line 18 – Page 12, Line 3; Page 14, Lines 8-24; and Figures 1 - 4).*

Claim 18 is similar to claim 4, and describes a computer program product tangibly embodied in a machine-readable medium (*e.g., memory 108*). The computer program product includes instructions for receiving executable code from a server system (*e.g., server 205 illustrated in Fig. 2*) in a client data processing system (*e.g., any of clients 215/220/225/230 illustrated in Fig. 2*). The computer program product also includes instructions for, if the client data processing system is in an idle state when the executable code is received (*e.g., at block 410*), sending a response to the server system (*e.g., at block 420*), testing at least a portion of the executable code (*e.g., at block 425*), and sending test results to the server system (*e.g., at block 430*). (*Application, Page 9, Line 15 – Page 10, Line 17; Page 10, Line 18 – Page 12, Line 3; Page 14, Lines 8-24; and Figures 1 - 4).*

Grounds of Rejection to be Reviewed on Appeal

1. Are Claims 8-14 indefinite under 35 U.S.C. § 112, second paragraph?

2. Are Claims 1-21 unpatentable under 35 U.S.C. § 103(a) over U.S. Patent Nos. 6,112,225 (Kraft *et al.*) and 6,360,268 (Silva *et al.*)?

ARGUMENT

Stated Grounds of Rejection

The rejections outstanding against the Claims are as follows:

1. In the March 21, 2008 Office Action, Claims 8-14 were rejected under 35 U.S.C. § 112, second paragraph, as indefinite for filing to particularly point out and distinctly claim the subject matter which applicant regards as his invention.

2. In the March 21, 2008 Office Action, Claims 1-21 were rejected under 35 U.S.C. § 103(a) as unpatentable over U.S. Patent Nos. 6,112,225 (Kraft *et al.*) and 6,360,268 (Silva *et al.*).

Legal Standards

There are two separate requirements under 35 U.S.C. § 112, second paragraph, according to MPEP § 2171. The first is subjective and requires that the claims must set forth the subject matter that the *Applicants* regard as their invention. The second is objective and requires that the claims must particularly point out and distinctively define the metes and bounds of the subject matter that will be protected by the patent grant (*i.e.*, whether the scope of the claim is clear to one of ordinary skill in the art). The Examiner should explain whether the rejection is based on indefiniteness or on the failure to claim what the Applicants regard as their invention. MPEP § 2171 *(citing Ex parte Ionescu*, 222 U.S.P.Q. 537, 539 (Bd. App. 1984)).

Obviousness: In rejecting claims under 35 U.S.C. § 103(a), the examiner bears the initial burden of establishing a *prima facie* case of obviousness. (*In re Oetiker*, 977 F.2d 1443, 1445, 24 USPQ2d 1443, 1444 (Fed. Cir. 1992). See also *In re Piasecki*; 745 F.2d 1468, 1472, 223 USPQ 785, 788 (Fed. Cir. 1984)). It is incumbent upon the examiner to establish a factual basis to support the legal conclusion of obviousness. (*Id.* at 1073, 5 USPQ2d at 1598). In so doing, the examiner is expected to make the factual determinations set forth in *Graham v. John Deere Co.*, 383 U.S. 1, 17, 148 USPQ 459, 467 (1966), *viz.*, (1) the scope and content of the prior art; (2) the differences between the prior art and the claims at issue; and (3) the level of ordinary skill in the art. In addition to these factual determinations, the examiner must also provide "some articulated reasoning with some rational underpinning to support the legal conclusion of obviousness." (*In re Kahn*, 441 F.3d 977, 988, 78 USPQ2d 1329, 1336 (Fed. Cir 2006) (cited with approval in *KSR Int'l v. Teleflex Inc.*, 127 S. Ct. 1727, 1741, 82 USPQ2d 1385, 1396 (2007)).

First Ground of Rejection

Claims 8-14 were rejected under 35 U.S.C. § 112, second paragraph, as indefinite for filing to particularly point out and distinctly claim the subject matter which applicant regards as his invention.

Claims 8-10

These claims may be considered together *for this ground of rejection.*

Independent claim 8 describes

> A data processing system comprising a processor and accessible memory, the data processing system configured to:
>
> receive a test request;
>
> send executable program code, corresponding to the test request, to a client system; and
>
> receive a response from the client system indicating that the client system will perform a test, and indicating that the client system was not being actively used when the executable program code was sent..

Several specific issues raised by the Examiner in the final Office Action were corrected in the (entered) after-final amendment, which added a colon and the word "comprising". These issues are believed obviated, and are not at issue in this appeal.

The Examiner also indicated that "it is unclear whether a data processing system merely being 'configured to' perform certain acts necessarily requires the inclusion of software encoding the acts such that the processor would perform the acts during execution of the software." Appellant respectfully notes that this is the very common way to claim a device that is particularly configured or adapted to perform certain functions, and that a generic general-purpose computer system cannot be said to be "configured" to perform the claimed functions.

Those of skill in the art are surely apprised of the claim scope, with no ambiguity at all. Those of skill in the art recognize that a special-purpose data processing system can be configured to perform specific functions either by executing specific software (as the Examiner notes), or by specifically configuring the processor to perform specific functions. A general-purpose computing system, lacking any software or particular configuration, will not perform such functions and cannot be said to be "configured" to perform those functions, as claimed and as recognized by those of skill in the art with no ambiguity or indefiniteness at all.

The indefiniteness rejections of these claims should be reversed.

Claims 11-14

These claims may be considered together *for this ground of rejection*. Independent claim 11 describes

> A data processing system comprising a processor and accessible memory, the data processing system configured to:
>
> executable code from a server system in a client data processing system; and,
>
> if the client data processing system is in an idle state when the executable code is received,
>
> send a response to the server system,
>
> test at least a portion of the executable code, and
>
> send test results to the server system.

Several specific issues raised by the Examiner in the final Office Action were corrected in the (entered) after-final amendment, which added a colon and the word "comprising". The Examiner is correct that the word "receive" is missing before the phrase "executable code" in the third line of this claim, above. This error was overlooked by Appellant in the after-final amendment, and Appellant will be happy to correct this error either by direct amendment or Examiner's amendment when proceedings on this appeal are concluded. Appellant does not appeal this specific rejection of claims 11-14 in their current form.

The arguments above with regard to "configured to" apply here as well, and are incorporated by reference.

Second Ground of Rejection

Claims 1-21 were rejected under 35 U.S.C. § 103(a) as unpatentable over U.S. Patent Nos. 6,112,225 (Kraft *et al.*, hereinafter "Kraft") and 6,360,268 (Silva *et al.*, hereinafter "Silva").

Claims 1-3, 8-10, and 15-17

These claims may be considered together *for this ground of rejection.*

Independent claim 1 describes

> A method for testing code, comprising:
>
> receiving a test request;
>
> sending executable program code, corresponding to the test request, to a client system;
>
> receiving a response from the client system indicating that the client system will perform a test, and indicating that the client system was not being actively used when the executable program code was sent.

Independent claim 8 is drawn to a data processing system particularly configured to perform similar functions as the method of claim 1, and independent claims 15 is drawn to a computer program product with instructions for performing similar functions as the method of claim 1.

Claim 1 requires receiving a test request. The Examiner argues that this is taught a Kraft at col. 9, lines 1-27. For the convenience of the Board, this portion of Kraft teaches:

When step 608 finds that the host peripheral computer 106 is idle, the screen saver 204 initiates its screen saver in step 609 to prevent damage to the peripheral computer's monitor; this is only an option, however, and the screen saver function may be omitted entirely. More important, the screen saver 204 in step 609 also sends a message to the task manager 206 indicating that the peripheral computer 106 is idle. In response, the task manger 206 in step 610 determines whether the task execution engine 208 is already engaged in computation of an uncompleted subtask. If so, the task manager 206 retrieves the intermediate results from the buffer 210 (step 616), and directs the task execution engine 208 to continue executing the subtask (step 618). In another embodiment, steps 610 and 616 may be omitted; in this case, uncompleted subtasks interrupted by the user's applications or other activity are aborted and restarted when the next idle period begins.

Alternatively, if step 610 finds no subtask already in progress, the task manager 206 in step 612 requests a subtask. This involves submitting a subtask request to the coordinating computer 102. To benefit the coordinating computer 102, the subtask request may be

> accompanied by a machine-readable description of the peripheral computer's hardware components, operating system, and the like. Upon receipt of the subtask (step 614), the peripheral computer 106 has "subscribed" to the coordinating computer's aggregate task. At this point, the task execution engine 208 may start computing the subtask in step 618.

Kraft clearly does not teach or suggest receiving a test request at all. Kraft does teach that a subtask request is sent to coordinating computer 102 (which, of course, receives it).

Claim 1 also requires sending executable program code, corresponding to the test request, to a client system. Kraft appears to teach that a subtask is then received by peripheral computer 106, which may have been sent by the coordinating computer 102.

Claim 1 also requires receiving a response from the client system *indicating that the client system will perform a test, and indicating that the client system was not being actively used when the executable program code was sent*. This is <u>not</u> taught or suggested by Kraft or by Silva; the Examiner does not allege any such teaching by Silva. Silva is only referenced for the concept of distributed testing.

For this limitation, the Examiner refers again to the passage reproduced above, but it is clear that Kraft's coordinating computer 102 does not receive any sort of response indicating that the peripheral computer will perform a subtask, nor does coordinating computer 102 receive any sort of response indicating that the peripheral computer was not being actively used when the subtask was sent.

The Examiner responds in the final Office Action by again referring to the col. 9 passage, and stating

> The system of Kraft on sends [*sic*] indications to the server (in the form of requesting new tasks and sending result of previous tasks) when it is otherwise not being actively used (see decision block 608 in Figure 6 (checking if the client system is idle)).

The Examiner is correct in that task requests and results are sent when the system is idle – but this is not the claim limitation. There is no response sent from the client to server indicating that the client <u>will</u> perform a task that it has been sent, as claimed – there appears to be no confirmation at all that a subtask has been received, or that the client will execute it. Nor is there any response sent from the client to server indicating that the client was not being actively used when the subtask was sent – it may have been idle when the task was originally requested, and it may be idle again at some point when the task is completed, but there is no indication that the client is idle when the subtask is received.

This can be important, of course, when the server must determine how soon a task is to be completed – in Kraft's system, there is no indication at all of when the task may be executed or completed. On the other hand, the claimed method provides that when the code is received by the client, it responds with an indication that it was idle on receipt and it will execute the task (perform the test). This is significantly different than the system of Kraft, and is not taught or suggested by Kraft.

The Examiner responds in the Advisory Action that there is no requirement of *when* various communications are sent, and that the claims only indicate an "eventual testing" that is confirmed when the test results are returned. This is a misreading of the plain language of the claims. The claims require a response that indicates that the client system *will perform* a test, clearly prospective language, and indicating that the client system *was not being actively used* when the executable program code was sent. The response clearly is sent after the program code is sent to the client, but before the test is performed.

Kraft does not teach or suggest this limitation of claim 1, or similar limitations of claims 8 and 15. Nor does Silva, nor does the Examiner allege any such teaching in Silva. As no art of record teach or suggest this limitation, no combination of the cited art can teach or suggest this feature. Nor is there any teaching or suggestion in the art to modify the references to meet this limitation, nor is there any teaching that would indicate that such a modification, if attempted would meet with predictable or successful results.

The obviousness rejections of these claims, and their dependents, should be reversed.

Claims 4, 6-7, 11, 13-14, and 18, 20-21

These claims may be considered together *for this ground of rejection.*

Independent claim 4 describes

> A method for testing code in a client data processing system, comprising:
>
> receiving executable code from a server system in a client data processing system;
>
> if the client data processing system is in an idle state when the executable code is received, then
>
> sending a response to the server system,
>
> testing at least a portion of the executable code, and
>
> sending test results to the server system.

Independent claim 11 is drawn to a data processing system particularly configured to perform similar functions as the method of claim 4, and independent claims 18 is drawn to a computer program product with instructions for performing similar functions as the method of claim 4.

The issues here are similar to those discussed above with regard to independent claims 1, 8, and 15. Claim 4 requires receiving executable code from a server system in a client data processing system. The Examiner again refers to Kraft's col. 9, lines 1-27, reproduced above. Kraft does teach that a "subtask" is received by peripheral computer 106. Though not specified by Kraft, it may be from coordinating computer 102.

Claim 4 also requires if the client data processing system is in an idle state when the executable code is received, then sending a response to the server system, testing at least a portion of the executable code, and sending test results to the server system. The Examiner again refers to Kraft's col. 9, but Kraft does not teach or suggest this limitation.

In Kraft's system, there is no response sent to the coordinating computer upon receipt of the subtask, whether or not the peripheral computer is idle when the subtask is received. Nor does Silva teach this limitation.

Contrast the limitation of Claim 5, discussed below, which requires that if the client data processing system is not in an idle state when the executable code is received, then no response is sent to the server and no testing is performed. This is also contrary to Kraft, which does not address the state of the client when the subtask is received, but will certainly execute the task and send a result whenever it does become idle.

None of the cited references teach or suggest this limitation of claim 4, alone or in combination, or the similar limitations of claims 11 and 18. The rejections, therefore, of claims 4-7, 11-14, and 18-21 should be reversed.

Claims 5, 12, and 19

These claims may be considered together *for this ground of rejection.*

Claim 5 requires that if the client data processing system is not in an idle state when the executable code is received, then no response is sent to the server and no testing is performed.

Claim 12 is drawn to a data processing system particularly configured to perform similar functions as the method of claim 5, and claims 19 is drawn to a computer program product with instructions for performing similar functions as the method of claim 5.

These claims depend from claims 4, 11, and 18, respectively, and so the arguments above with respect to those claims apply here as well, and are hereby incorporated by reference.

These claims require that if the client data processing system is not in an idle state when the executable code is received, then no response is sent to the server and no testing is performed. Neither Kraft nor Silva discuss at all what happens if the client system is not in an idle state *when the executable code is received*, and they certainly don't teach this feature.

The Examiner refers to Kraft's col. 9, lines 36-55, which describe:

> Interrupt Handling
>
> As mentioned above, subtask computation occurs in the peripheral computers 106 during "idle" processing times. When the peripheral computer 106 is active (i.e., not "idle"), work toward completing the subtask is suppressed. During these times, the peripheral computer 106 performs normal tasks as configured by its operator. FIG. 7 illustrates one example of this process in some detail as shown by the steps 700. In this example, when the peripheral computer 106 becomes active, it generates a hardware interrupt, as shown by step 702. In response, the screen saver 204 deactivates its screen saver function. Accordingly, the peripheral computer's monitor is powered up (if power has been removed), and permitted to display the image being output by the peripheral computer's processor. As another consequence of the interrupt of step 702, the task execution engine

> 208 pauses its subtask computation in step 706. The task execution engine 208 saves its intermediate computations in the buffer 210. After steps 704-708 complete, the interrupt routine 700 ends in step 710.

Note that this passage only describes that subtask computation is "suppressed" when the processor is not idle. This passage does not address what happens *when the executable code is received*, as claimed.

On the contrary, Kraft specifically describes that the subtask is requested and received by the peripheral computer *when it is idle*, in the passage from col. 9 reproduced above with regard to claim 1. Kraft teaches that "When step 608 finds that the host peripheral computer 106 is idle, ... the screen saver 204 in step 609 also sends a message to the task manager 206 indicating that the peripheral computer 106 is idle. In response, the task manger 206 in step 610 determines whether the task execution engine 208 is already engaged in computation of an uncompleted subtask. ... if step 610 finds no subtask already in progress, the task manager 206 in step 612 requests a subtask. This involves submitting a subtask request to the coordinating computer 102. ... Upon receipt of the subtask (step 614), ... the task execution engine 208 may start computing the subtask in step 618" (passage in full is above; this is edited to the relevant description).

It is clear that the subtask is requested and received by the peripheral computer *when it is idle*, and that the peripheral computer may start executing it immediately. The interrupt handling section reproduced above indicates that this processing is *paused* when the system is not idle, but

Kraft clearly describes that after the pause, the subtask execution resumes at the next idle time, and the computation results are eventually sent to the coordinating computer.

Kraft does not contemplate, teach, or suggest, that if the client data processing system is not in an idle state when the executable code is received, then no response is sent to the server and no testing is performed. On the contrary, Kraft's system only requests tasks when it is idle, and all received tasks are eventually executed with results being sent to the server. This is opposite that which is required by claims 5, 12, and 19.

The rejections of these claims should be reversed.

All obviousness rejections should be reversed.

Grouping of Claims

The claims on appeal do not stand or fall together, as may be seen from the arguments set forth below. Each claim or group of claims that has been argued separately under a separate subheading should be considered separately. While the appellant recognizes that a formal statement regarding the grouping of claims is no longer required, each claim should be considered separately; or at the very least each claim which is argued separately in the preceding sections of this brief should be considered separately.

REQUESTED RELIEF

The Board is respectfully requested to reverse the outstanding rejections and return this application to the Examiner for allowance.

Respectfully submitted,

MUNCK CARTER, P.C.

Date: 09/22/08

Matthew S. Anderson
Registration No. 39,093

P.O. Drawer 800889
Dallas, Texas 75380
(972) 628-3600 (main number)
(972) 628-3616 (fax)
E-mail: *manderson@munckcarter.com*

ATTORNEY FOR APPELLANT

DOCKET NO.: 05-03-010 (UGSC01-05022) **PATENT**

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of	:	Graham Hughes, et al.
Serial No.	:	10/706,848
Filed	:	November 12, 2003
For	:	SYSTEM, METHOD, AND COMPUTER PROGRAM PRODUCT FOR DISTRIBUTED TESTING OF PROGRAM CODE
Group No.	:	2192
Examiner	:	Eric B. Kiss
Conf. No.	:	6082

APPENDIX A -

Claims Appendix

1. (Original) A method for testing code, comprising:

receiving a test request;

sending executable program code, corresponding to the test request, to a client system;

receiving a response from the client system indicating that the client system will perform a test, and indicating that the client system was not being actively used when the executable program code was sent.

2. (Original) The method of claim 1, wherein executable program code, corresponding to the test request, is sent to multiple client systems.

3. (Original) The method of claim 1, further comprising retrieving a list of client system identifiers, the client system identifiers indicating client systems to which executable program code can be sent for testing.

4. (Original) A method for testing code in a client data processing system, comprising:

receiving executable code from a server system in a client data processing system;

if the client data processing system is in an idle state when the executable code is received, then

sending a response to the server system,

testing at least a portion of the executable code, and

sending test results to the server system.

5. (Original) The method of claim 4, wherein if the client data processing system is not in an idle state when the executable code is received, then no response is sent to the server and no testing is performed.

6. (Original) The method of claim 4, wherein the testing is a coverage analysis test.

7. (Original) The method of claim 4, wherein the client data processing system is in an idle state when no user is actively operating the client data processing system.

8. (Previously Presented) A data processing system comprising a processor and accessible memory, the data processing system configured to:

receive a test request;

send executable program code, corresponding to the test request, to a client system; and

receive a response from the client system indicating that the client system will perform a test, and indicating that the client system was not being actively used when the executable program code was sent.

9. (Original) The data processing system of claim 8, wherein executable program code, corresponding to the test request, is sent to multiple client systems.

10. (Previously Presented) The data processing system of claim 8, wherein the data processing system is further configured to retrieve a list of client system identifiers, the client system identifiers indicating client systems to which executable program code can be sent for testing.

11. (Previously Presented) A data processing system comprising a processor and accessible memory, the data processing system configured to:

executable code from a server system in a client data processing system; and,

if the client data processing system is in an idle state when the executable code is received,

send a response to the server system,

test at least a portion of the executable code, and

send test results to the server system.

12. (Original) The data processing system of claim 11, wherein if the client data processing system is not in an idle state when the executable code is received, then no response is sent to the server and no testing is performed.

13. (Original) The data processing system of claim 11, wherein the testing is a coverage analysis test.

14. (Original) The data processing system of claim 11, wherein the client data processing system is in an idle state when no user is actively operating the client data processing system.

15. (Original) A computer program product tangibly embodied in a machine-readable medium, comprising:

instructions for receiving a test request;

instructions for sending executable program code, corresponding to the test request, to a client system;

instructions for receiving a response from the client system indicating that the client system will perform a test, and indicating that the client system was not being actively used when the executable program code was sent.

16. (Original) The computer program product of claim 15, wherein executable program code, corresponding to the test request, is sent to multiple client systems.

17. (Original) The computer program product of claim 15, further comprising instructions for retrieving a list of client system identifiers, the client system identifiers indicating client systems to which executable program code can be sent for testing.

18. (Original) A computer program product tangibly embodied in a machine-readable medium, comprising:

instructions for receiving executable code from a server system in a client data processing system;

instructions for, if the client data processing system is in an idle state when the executable code is received,

sending a response to the server system,

testing at least a portion of the executable code, and

sending test results to the server system.

19. (Original) The computer program product of claim 18, wherein if the client data processing system is not in an idle state when the executable code is received, then no response is sent to the server and no testing is performed.

20. (Original) The computer program product of claim 18, wherein the testing is a coverage analysis test.

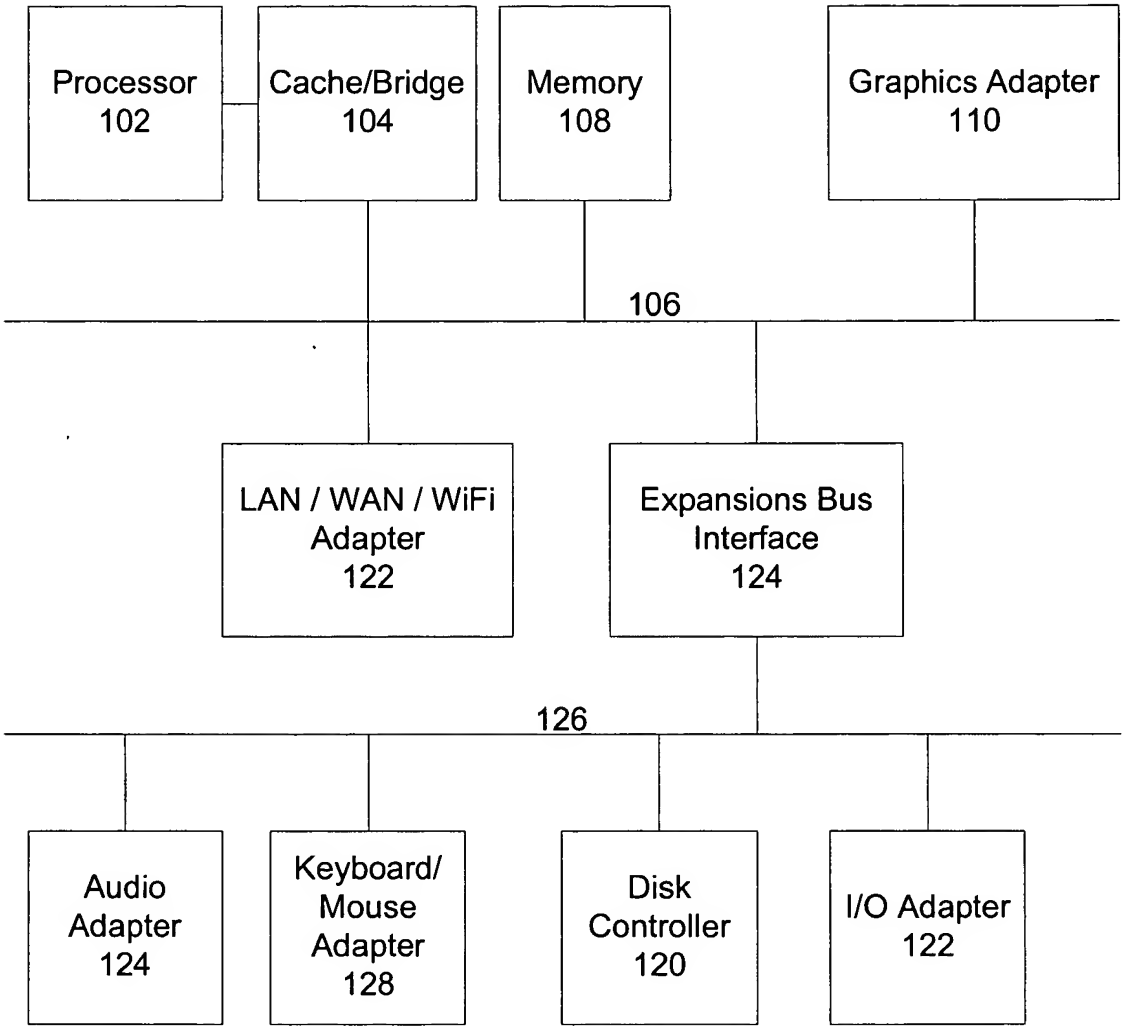
21. (Original) The computer program product of claim 18, wherein the client data processing system is in an idle state when no user is actively operating the client data processing system.

DOCKET NO.: 05-03-010 (UGSC01-05022) PATENT

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of	:	Graham Hughes, et al.
Serial No.	:	10/706,848
Filed	:	November 12, 2003
For	:	SYSTEM, METHOD, AND COMPUTER PROGRAM PRODUCT FOR DISTRIBUTED TESTING OF PROGRAM CODE
Group No.	:	2192
Examiner	:	Eric B. Kiss
Conf. No.	:	6082

APPENDIX B -
Copy of Formal Drawings



PRIOR ART

Figure 1

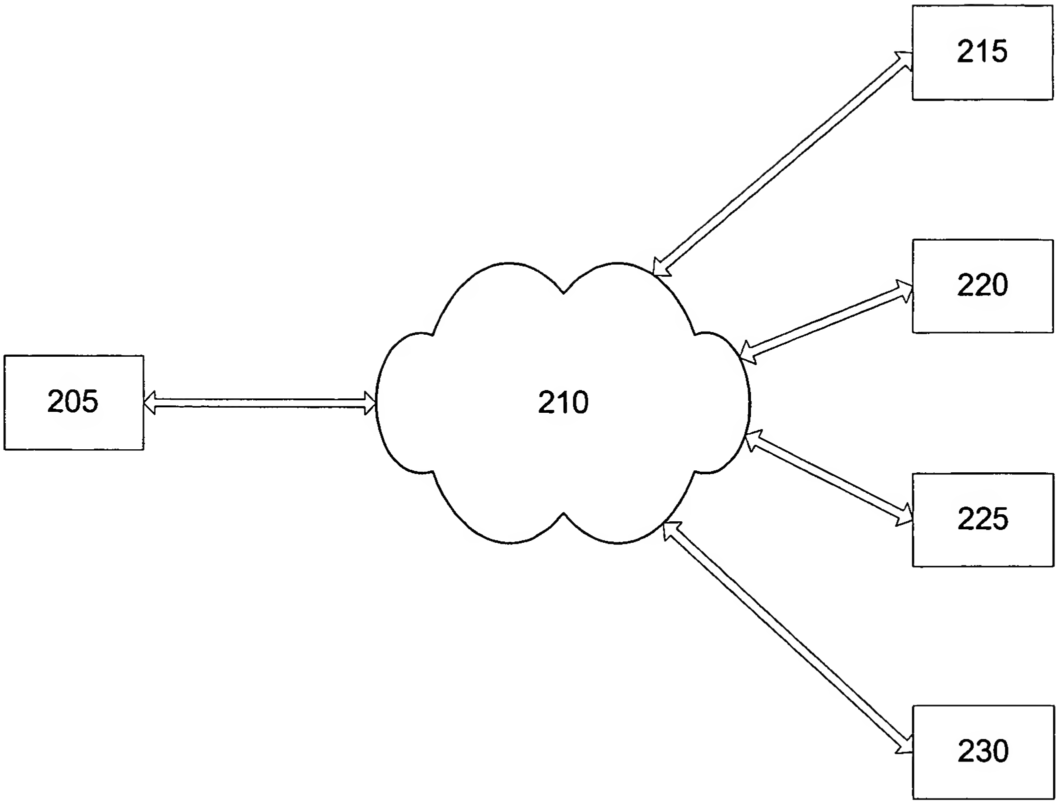


Figure 2

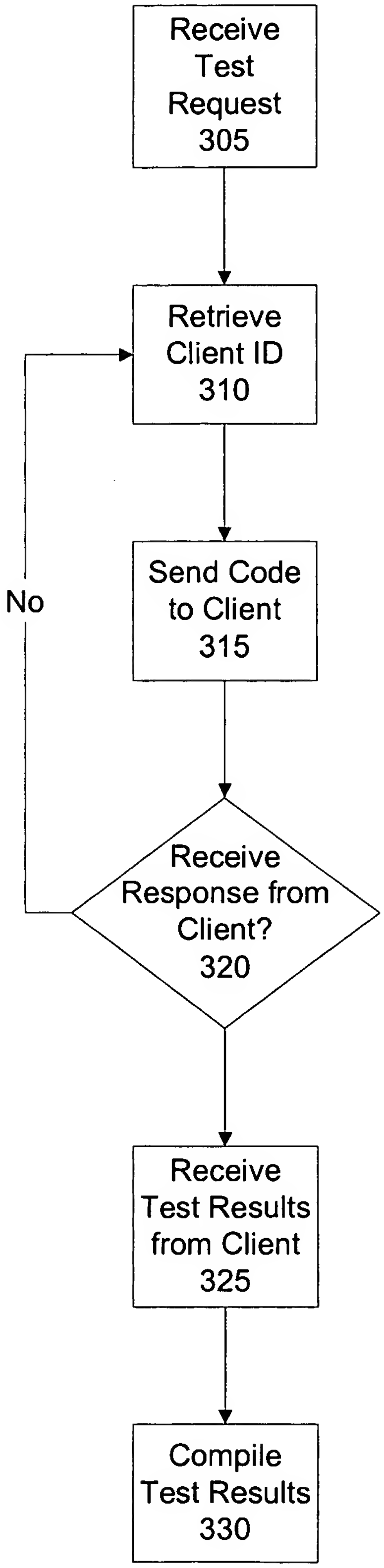

Receive Test Request 305
Retrieve Client ID 310
Send Code to Client 315
No
Receive Response from Client? 320
Receive Test Results from Client 325
Compile Test Results 330

Figure 3

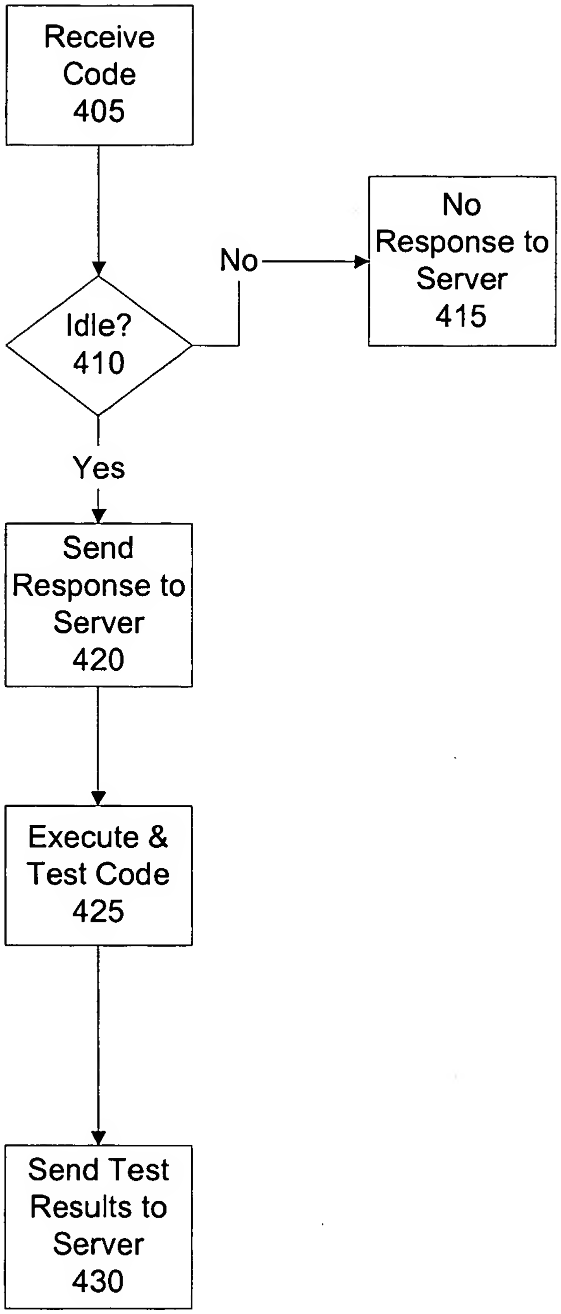

Receive Code 405
Idle? 410
No
No Response to Server 415
Yes
Send Response to Server 420
Execute & Test Code 425
Send Test Results to Server 430

Figure 4

DOCKET NO.: 05-03-010 (UGSC01-05022) **PATENT**

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of	:	Graham Hughes, et al.
Serial No.	:	10/706,848
Filed	:	November 12, 2003
For	:	SYSTEM, METHOD, AND COMPUTER PROGRAM PRODUCT FOR DISTRIBUTED TESTING OF PROGRAM CODE
Group No.	:	2192
Examiner	:	Eric B. Kiss
Conf. No.	:	6082

APPENDIX C -
Evidence Appendix

Not Applicable -- No other evidence was entered.

DOCKET NO.: 05-03-010 (UGSC01-05022) **PATENT**

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of	:	Graham Hughes, et al.
Serial No.	:	10/706,848
Filed	:	November 12, 2003
For	:	SYSTEM, METHOD, AND COMPUTER PROGRAM PRODUCT FOR DISTRIBUTED TESTING OF PROGRAM CODE
Group No.	:	2192
Examiner	:	Eric B. Kiss
Conf. No.	:	6082

APPENDIX D -

Related Proceedings Appendix

Not Applicable -- To the best knowledge and belief of the undersigned attorney, there are none.